Coarsening Algorithm for Tetrahedral Meshes

Student: Eve Schnebly Crow

Faculty Mentor: Steven Benzley, Civil and Environmental Engineering

In Finite Element Analysis (FEA), there are often times when the mesh on the model needs to be more dense or less dense in a certain region. Refinement and coarsening are the tools used to achieve the desired density of a mesh. The CUBIT[1] toolkit has a coarsening algorithm for triangle surface meshes, but the algorithm for coarsening tetrahedral meshes has not been implemented. I have done some research and have a plan for how it could be implemented.

CUBIT uses the third party library Tetmesh-GHS3D, developed by Inria laboratory in France, for it's tetrahedral meshing[2]. One option for localized coarsening of tetrahedral meshes is to use an interface with Inria's meshing tool. The desired region for coarsening would be selected by the user or by a future automated process. This region would be wrapped in a temporary surface or boundary. All of the tetrahedral elements inside the desired region would be deleted. The temporary surface would be coarsened using the existing triangle-coarsening algorithm. Then the internal region would be remeshed with Inria's meshing tool with the desired size. The temporary surface would then be deleted except for any part that coincides with a surface on the model. This would be a quick way to implement coarsening a portion of the tetrahedral mesh.

One problem with the above-mentioned method of tetrahedral coarsening is the robustness of the triangle-coarsening algorithm. I have reworked this algorithm to be more robust, but it still needs more work in order to be used for tetrahedral-coarsening. Right now it will not coarsen a triangle surface mesh that has tetrahedral elements attached to it. This is essential for the proposed method of tetrahedral coarsening in order to maintain a conformal mesh[3].

Another problem is the flexibility of controls on the density of the mesh around the selected region. The current localized refinement algorithms have options to control the number of times to refine the region, the size of elements with or without a bias on how quickly to return to the original size in the area around the selected region, and depth or radius to refine from region into existing mesh. The control options for coarsening should be similar to these options. Most of these already exist for the triangle-coarsening algorithm. The proposed method for tetrahedral-coarsening would not have as much flexibility to control the transition region as it would if a method similar to the triangle method were used or if several sub-regions were coarsened to give the desired level of overall coarsening.

The other option for tetrahedral-coarsening that I have researched would be much more time consuming to implement. It is to use an algorithm that is similar to the one used for triangles, but expanded to 3-D elements. Expanding to the third dimension is more difficult than it might seem. It is more difficult to get a good quality coarsened tetrahedral mesh than it is for a triangle mesh. This means there would have to be more checks and corrections to ensure a

---

[1] CUBIT is an full-featured software toolkit for robust generation of two- and three-dimensional finite element meshes (grids) and geometry preparation. <http://cubit.sandia.gov/release_notes/10.0.html>

[2] <http://cubit.sandia.gov/release_notes/10.0.html#tet_upgrade>

[3] Conformal mesh is where the edges of all the elements line up.

good mesh. It also means that it would take longer to write and also to run once it is implemented. One of the goals for CUBIT algorithms is to make them as fast and efficient as possible.

Considering the pros and cons of each of these methods the first option was selected. That is to use an interface with Inria's meshing tool and the existing triangle-coarsening algorithm. Additional work will need to be done to improve the triangle-coarsening algorithm and determine how to implement the control options for tetrahedral-coarsening.

I was not able to completely implement tetrahedral-coarsening into the CUBIT toolkit because of the additional work that had to be done on improving the triangle-coarsening algorithm and because of other higher priority items that came up on the CUBIT project.